Prabodh
PAGE No. :
DATE :

what is C++ ?

- C++ is a statically typed, free form, (usually) compiled, multi paradigm, intermediate level general purpose middle level programming language.

statically type → we have to follow a confined syntax.

free form → we can write in any form just follow the syntax.

multi paradigm → more than one method to solve.

intermediate → not even a machine code & also not even a common english

level.

latest version of C++ is C++14

- C++ is a new version of C. developed in 1979, By. Bjarne Stroustrup.

- wherever there is work of graphics there is C++ many of today's operating system, system drivers, browsers, games they use C++ as their core language, this make C++ meaning. today

C & C++ are similar

↳ it contains everything of C++

why c++ ?

Prabodh
PAGE No. :
DATE :

- c++ is irreplaceable. because it is light when it comes to storage. and also it is used as core language. to deal with hardware it is necessary to give light store.
- c++ is very simple and by using this we can learn the internal architecture of a computer.

Features of c++.

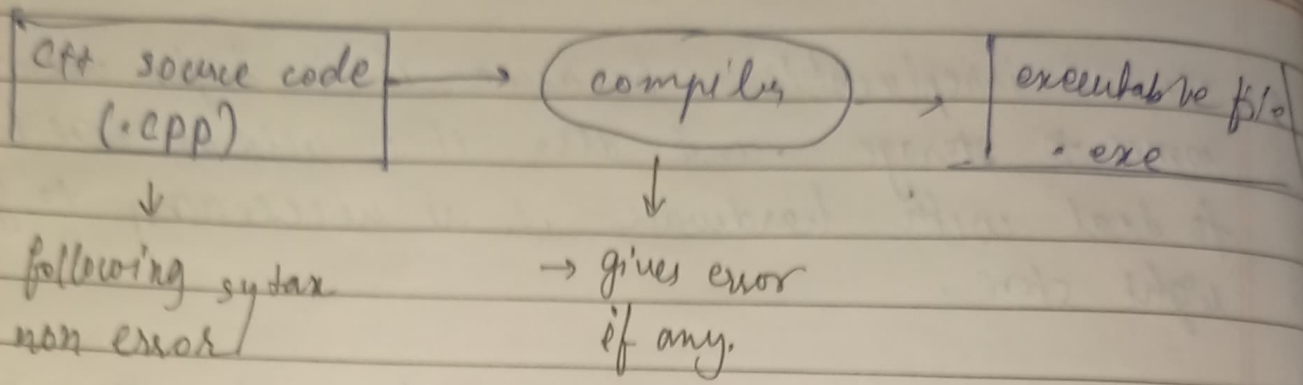
- c++ is fast. light programming language.
- uses syntax
- multi paradigm programming language. you can have more than one method.
- it is object oriented programming but not completely like Java.
- Power of standard library. (Standard template library STL)

where do we find c++ ?

- adobe illustrator
- photoshop
- any graphical work
- facebook
- amazon storage / web service / Data base service
- Bitcoin.

Prabodh
PAGE No.:
DATE:

what C++ compiler does (Basic)



C++ tokens ←

- identifies.
- keywords
- operators
- constant / literals

identifies → to identify thing.

A C++ identifies is a name given to variable function, class module or any user defined item.

rules → identifies start with letter A to Z or a to z or (_) underscore
→ followed by many A to Z or a to z or 0 to 9
→ C++ does not allow punctuation @. $ or %
→ it is case sensitive.

Prabodh
PAGE No.:
DATE:

Keywords:-

in C++ you cant use keyword as identifier as they are reserved in C++ library to perform an internal operation.

Operators:-

an operator is a symbol that tell the compiler to perform specific mathematicle or mathematical operation.

- arithmetic → like (+)(-)(÷)(%) etc. variable or letter (+) to get sum.
- relational → relation between numbers like >< ≠ etc.
- logical → it give logical like in boolean 1⊕1 = 1 and.
- Bit wise
- assignment → to assign a value in a variable.
- miscllenous

constant.
using const. keyword we can make them constant.

Data Types —
there are four types of C++

- Basic — int, float, char, double
- Derived - array, string, pointers
- Enumeration - enum
- user difined - struct, class

int → integer, float → rational. decimal (only store upto 4 decimal place

char → character (a, b, c etc)

double → more range than float

Prabodh
PAGE No. :
DATE :

array → to save continous data
for integers int. array
for characters char array.

Basic program.

```
#include <iostream>
using namespace std;

int main()
{
    cout << "hello world!";
    return 0;
}
```

execution

output → hello world!

#include <iostream> → input output stream

↓ its pre processor to include this for no error in forward

cout → (c) out

↓

for ctr

using namespace std;

↓

to tell it has to use standard output

int main → to start function.

↓

section of code

Prabodh
PAGE No. :
DATE :

semi colon used for ending the line for that line of code.

row.

```
#include <iostream>
using namespace std;
```
} compulsory

```
int main ()
{
    int a;
    int b;
    int q;
```
→ int a, b, q;

```
    float f;
    char c;
    double d;
    cout<<"hello world"
    return 0;
}
```
finished with this

now here we can say a, b, c, q, f, d are like containes which can only store single ~~dt~~ data.

a, b, q → only store integer type data
f → only store floating type data
c → only store character type data
d → only store double (more decimal data)

→ they can vary.

Prabodh
PAGE No. :
DATE :

now how to define value there are two ways.

```
int main()
{
    int a, b, q;        or  int q=12
    float f;
    char c;                  both can work
    double d;
    q=12;
    cout<<"hello world"
    return 0;
}
```

now if we put q=13 in bottom

```
int main()
{
    int q=12, b, q;
    float f;                  here q=12 which is
    char C;                        stored.
    double d;
    q=13;          ---------> but here it becomes 13.
```

here (=) equal to is assignment operator, it assigns values

```
int main()
{
    int q=12, b=10;
    float f=3.5;
    char c='A';
    double d=3.56798;
```

Prabodh
PAGE No.
DATE

in C++ code execution starts from back.

now let is see for z value

```
int main()
{
    int a=12, b=10, z;
    float f=3.5;
    char c='A';
    double d=3.56778;
    z=a+b
```

→ now sum of a and b will be assigned to z.
common maths. but.

if we we a=a+b let see what happened.

→ now this value will assigned to a.

now to print all this things let's see,
so things inside quotes (" ") can be print directly but to print variable we just need to write its variable name

```
int main()
{
    int a=12, b=10, z;
    float f=3.5;
    char c='A';
    double d= 3.56778;
    cout<<a;
    a=a+b;
    cout<<a;
    return 0;
}
```

output = 1222

Prabodh
PAGE No. :
DATE :

now to get space between them we just need to give a blank quot

```
cout<<a<<" ";
```

it give 12 22

to get next line we use special character which is already assigned in C++ which is \n.

```
cout<<a<<"\n"
```

output → 12
22

another method → cout<<a.<<endl;

↓
end of line without quote write that.

output → 12
22

now to print other thing with space we give

```
cout<<a<<" "<<b<<" "<<f;
```

output → 12
22 10 3.5

Condition Statement

Prabodh
PAGE NO.:
DATE:

- Ternery operation
  given condition and two statements if statement one doesnot give ~~requ~~ true result then it will give statement 2.
- If-else
- If-else if-else
- nested if-else

(1) Ternery operator.

```
#include <iostream>
using namespace std;

int main()
{
    int i=15;
    i>10 ? cout<<"greater" : cout<<"smaller";
    return 0;
}
```

output : Greater

now if we use 5 in place of 15

```
#include <iostream>
using namespace std;
int main()
{
    int i=5;
    i>10 ? cout<<"greater" : cout<<"smaller";
    return 0;
```

output : smaller.

Prabodh
PAGE No. :
DATE :

If - else

```
#include <iostream>
using namespace std;

int main()
{
  int i = 15;
  if(i>10)
    cout << "greater";
  else
    cout << "smaller";
  return 0;
}
```

output → greate

now if we use 5 instead of 15

```
#include <iostream>
using namespace std;

int main()
{
  int i=5
  if (i>10)
    cout << "greater";
  else
    cout << "smaller";
  return 0;
}
```

out put: smaller

if ~~then~~ statement is wrong and there is no else statement than there will a blank or no output.

Prabodh
PAGE No.:
DATE:

* if-else if-else

```
#include <iostream>
using namespace std;

int main ()
{
  int i = 25;
  if (i>10)
  {
    cout << "greater than 10";
  }
  else if (i > 20)
  {
    cout << "greater than 20";
  }
  else
  {
    cout<<"greater";
  }
    return 0;
}
```

we have to remove this else by removing this our compiler will go through this also which gives output

and also we "\n greater than 20" by this we will get that in new line.

out put : greater than 10

but here it is also greater than 20 which should be printed. But it only print if statements if they are true if we want print second line also they we have to we don't string

out put : greater than 10
greater than 20

Prabodh
PAGE No.:
DATE:

For - Loop

Types of loop

- while loop
- do while loop
- for loop
- infinite loop
- Break & continue statements

for loop

```
#Include <iostream>
using namespace std;

int main ()
{
   for (int i=0; i<10; i++)
   {
      cout<< i <<" ";
   }
   return 0;
}
```

differentiate between commands

for increment

given integer

it will first check whether 0 ei not less than 10 or not then if its true then it will be printed and we also given a space. now after this due to we used i++ this for increment. after that it will check 1 is less than 10 or not which is true it will print till 9 because statement is true till the

output :- 0 1 2 3 4 5 6 7 8 9

Prabhat
PAGE No.:
DATE:

To print first even numbers

```
#include <iostream>
using namespace std;

int main()

{ for (int i=0 ; i<10 ; i++)
  {
    if ((i % 2)==0)
    {
      cout<<i<<" ";
    }
  return 0;
}
```

→ mod, it will give reminder.

output → 2 4 6 8 10

To print odd numbers

we just have to put if((i%2)!=0) → if is used for opposite like here if means it shouldn't be divisible 2

To print table of twos

```
#include <iostream>
using namespace std;
int main()
{
  for (int i=1 ; i<=10 ; i++)
  {
    cout<<i*2<<" ";
  }
```

out- 2 4 6 8 10 12 14 16 18 20

Q1. Print nth term of an A.P and sum of nth term.

Prabodh
PAGE NO :
DATE :

a = 5 , d = 2

n^{th} term = a + (n-1) x d

sum of n^{th} term ⇒ $S = \frac{n}{2}(2a + (n-1)d)$

```
#include <iostream>
using namespace std;
int main()
{
   int a=5, d=2, an=0, sn=0, n, i;
   cin >> n ;
   for (int i=1; i<=n; i++)
   {
      an = a + (i-1) * d;
      sn = an + sn;
   }
   cout << "term" << " " << n << " = " << an << endl;
   cout << "sum of nth term = " << sn << endl;

   return 0;
}
```

nth term → an
sum of nth term → sn
n → input
i → input, just because it is integer type.

if we use for loop we have to use this without for loop just use formula of Sum of AP.

output : input 3
term 3 = 9
sum of nth term = 21

an

To find factorial of a given number

```
#include<iostream>
using namespace std;
int main()
{
   int num, fac=1;
   cout<<"enter yours no. to get factorial";
   cin>>num
   for(int a=1;a<=num;a++)
   {
     fac=fac*a;
   }
cout<<"factorial of given no. is"<<a<<fac<<endl;
  return 0;
}
```

Easy

bytes

output : enter yours no. to get factorial = (5)
factorial of a given no. is = 120.

```
#include <iostream>
using namespace std;

int factorial (int n)
{
   int ans=1
   for(int i=n;;i>0;i--)

      ans = ans *i ; or ans=*i;

      return ans;
}
int main()
{     int no;
   cout<<"enter no. of factorial";
```

(1) more important

Same but with function

```
cin >> no ;
cout<<no<<"! = " << factorial (no) << endl;
                                    return 0;
                                }
```

To check prime number

```
#include <iostream>
using namespace std;

int main;
{
   int n,i;
cout-<<" enter no. to check prime";
cin >> n ;

for (int i=2 ; i<n ; i++)
{
   if(n%i)==0 )
   {  cout<<i<<endl;
      cout<<"not prime"<< endl;
         break;
   }
}
  if (i==n)
{
   cout<<" prin is a prime number"; end
}

   return 0;
}
```

→ to check when i'm get out

now we have to reduce ↑

```
i < n/2

if (i==n/2)
```

Programming

i*i i+i<2 i=2 i<n i<√n

Prabodh
PAGE No.:
DATE:

to reduce more we look.

Fibonacci Series :-

first of all understand fibonacci series

fibonacci series start from 0 and we have to add previous two numbers to get next num

0 1 1 2 3 5 8 13 21 34 55 .....

important

coding ↓

```
#include <iostream>
using namespace std;
int main()
{
    int n, a=0, b=1, c;
    cout<<"enter no. of term for fibonacci seres";
    cin>>n; cout<<a<<" "<<b<<" ";
    for(int i=2; i<n; i++)
    {
        c=a+b;
        cout<<c<<" ";
        a=b;
        b=c;
    }
    cout<<endl;
    return 0;
}
```

Prabodh
PAGE No.
DATE

Fibonacci series by using arrays

See online Gdb.

Function :-

First we study about for pattern.

(pattern)

need to study discuss ideas

```
#include <iostream>
using namespace std;

void pattern()
{
    int n;
    cout << "enter no. of rows";
    cin >> n;

for (int i=0; i<n; i++)
    {
        for (int j=0; j<=i; j++)
        {
            cout << "*";
        }
            cout endl;
        }

}

int main()
{
    pattern();
    pattern();
    pattern();
return 0;
```

Programming fundamentals — Call by value

Prabodh
PAGE No. :
DATE :

sending values (pass by value)

To get exponential values ()

```
#include <iostream>
using namespace std;

int pow(int no, int po)
{
    int ans = 1;
    for (int i=0; i<po; i++)
    {
        ans = ans * no;
    }
    return ans;
}

int main()
{
    int n, p, answer;
    cout << "enter no. :";
    cin >> n
    cout "enter powe :";
    cin >> p

cout << "answer is :" << pow(n,p) << endl;
return 0;
}
```

for recursive function check DS Notebook

Programming fundamental using C++

Introduction to C++ and C

Prabodh
PAGE No. :
DATE :

```
#include <iostream>
using namespace std;

float division ( float dividend, float divisor )
{
    float ans = dividend / divisor
    return ans;
}

int main()
{
    float dividend, divisor;
    cout << "Enter dividend:";
    cin >> dividend;
    cout << "Enter divisor:";
    cin >> divisor;
    cout << dividend << "/" << divisor << "
    division ( dividend, divisor ) << endl;
    return 0;
}
```

ASCII table.

Prabodh
PAGE No.:
DATE:

Character function: it will return character whether in given input which alphabet is capital.

code :

```
#include <iostream>
using name space namespace std;

char firstcap (char arr[10], int n)
{
    for (int i=0; i<n; i++)
    {
        if (arr[i] >= 'A' && arr[i] <= 'Z')
            return arr[i]
    }
}
int main ()
{
    int n=10;
    char arr[n];

    for (int i = 0; i<n ; i++)
    cin >> arr[i];
    cout<< " first capital letter is " << firstcap
    (arr, n) << end;
    return 0;
}
```

Programming fundamental using C++

(true/false)

Prabodh
PAGE No.:
DATE:

Bool function

→ check given prime number is prime or not

```
#include <iostream>
using namespace std;
bool isprime(int n)
{
    if (n<=2)
    { return true;
    }
    for (int i=2; i*i<=n; i++)
    { if((n%i)==0)

        return false;
    }
    return true;
}
int main()
{
    int no;
    cout<<"enter no. to becheeked:";
    cin>>no;

    if(isprime(no))
    {
        cout<<no<<" is a prime number"
        endl;
    }

    else
    {
        cout<<no<<" is not a prime
        number"<<endl;
    }
```

Prabodh
PAGE No. :
DATE :

```
    return 0;
}
```

Output → GDB.

type of function for passing value :-

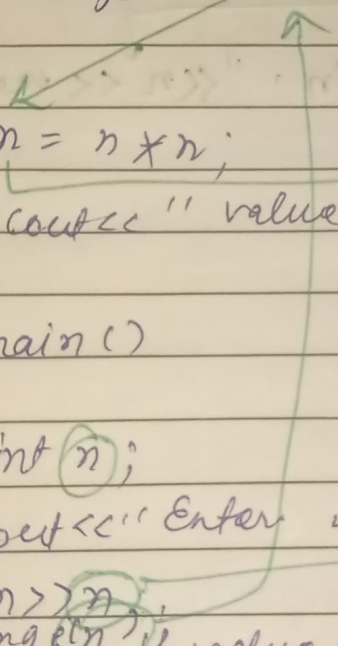
Call by reference & Call by value

program :→

(call by value)

```
#include <iostream>
using namespace std;

void change (int n)
{
    n = n * n;
    cout << " value in function : " << n << endl;
}
int main()
{
    int n;
    cout << " Enter value : ";
    cin >> n;
    change (n);
    cout << " value in main ; " << n << endl;
    return 0;
}
```



out put → GDB

Programming fundamental using C++

Prabodh
PAGE No. :
DATE :

call by reference

```
#include <iostream>
using namespace std;
void change(int &n)
{
    n = n*n;
    cout<<" value in function : "<< n <<endl
}
int main()
{
    int n;
    cout << "Enter value : ";
    cin>> n;
    change(n);
    cout<< value in main : "<<n << endl;
    return 0;
}
```

new pointer not new value

output → GDB

1 = true
0 = false

Prabodh
PAGE No.:
DATE:

(C++) advanced pattern:

While Loop

```
#include <iostream>
using name space std;
int main()
{
   while(1)
    {
       cout << "hi";
    }
     return 0;
}
```

Output → infinite hi

```
#include <iostream>
using namespace std;
int main()
{
   int a=0;
   while (a<10)
    {
       cout<<"hi" << endl;
       a++;
    }
    return 0;
}
```

out put → GDB

Prabodh
PAGE No.:
DATE:

more than 1 statement

```
#include <iostream>
using namespace std;
int main()
{
    int a=6;
    while (a<10 && a>5)
    {
      cout<<"hi"<<endl;
      a++;
    }
    return 0;
}
```

&& - and-and.
logical operator
it will check both condition
if either one is not true than it won't run.

output =
hi (6)
hi (7)
hi (8)
hi (9)

```
#include <iostream>
using namespace std;
int main()
{
    int a=4;
    while (a<10 || a>5)
    {
      cout<<"hi"<<endl;
      a++;
    }
    return 0;
}
```

|| - or-or
again logical operator
it will check any one if only one is true than it will run

output → GDB but here again it become infinite because at any

point they will become true.

Do-while loop

```
#include <iostream>
using namespace std;
int main()
{
    int a=4;
    while (a>4)
    {
     cout<<"hi"<<" ";
     a++;
    }
      return 0;
}
```

print → nothing will be printed

```
#include <iostream>
using namespace std
int main()
{
  int a=4;
  do
  {
  cout<<"hi"<<endl
    a++;
  }
  while(a>4);
  return 0;
}
```

print → infinite (hi) because here it will become 5

```
#include <iostream>
using namespace std;
int main()
{
  int a=4;
  do
   {
    cout<<"hi"<<endl;
    a++;
   }
   while(a>4 && a<10);
   return 0;
}
```

print same as 1st

in for loop leave blank

```
for(;;)
```

Break & Continue Statement

```
#include <iostream>
using namespace std;
int main()
{
   int a=0;
   while(a<10)
   {
      if(a==5)
         continue;
      cout<<a<<endl;
      a++;
   }
   return 0;
}
```

output: 1
2
3
4

```
#include <iostream>
using namespace std;
int main()
{
    int a=0;
    while (a<10)
    {
        a++;
        if(a==5)
        continue;
        cout<<"a<<endl;
    }
    return 0;
}
```

output :

```
1
2
3
4
6
7
8

9
```

break;

```
1
2
3
4
```

Data types

C++
data types

user defined	derived	primitive
structure	array	integer
union	function	char
enum	Pointer	float
		double
		void
		bool.

built-in data types

let we talk about integers can hold non decimal values 26, 373, -1729.
normally integers can store -32768 to 32767

however we can use
short
long
unsigned
} int

for
for -2,147,483,648
for 0 to 65536
only (+)

floating type data (float)
it can store decimal value.

Char data type store character -

variable	keyword	bytes	Range
Character	char	1	-128 to 127
unsigned char	unsigned char	1	0 to 255
integer	int	2	-32768 to 32767
Short integer	short int	2	—11—
long integer	long int	4	-2,147,483,648 to (+)
unsigned integer	unsigned int	2	0 to 65535
unsigned long —11—	unsigned ~~short~~ long int	2	—11—
unsigned short —11—	unsigned short int	4	
float	float	4	
double	double	8	
long double	long double	10	

Variable

a variable is most fundamental aspect of any computer language. It is a location in computer memory which can store data in symbolic name

declaration of variables

example
```
int a;
float mynumber;
```

if you want to declare more then one variable of same type you can use them in single statement with comma

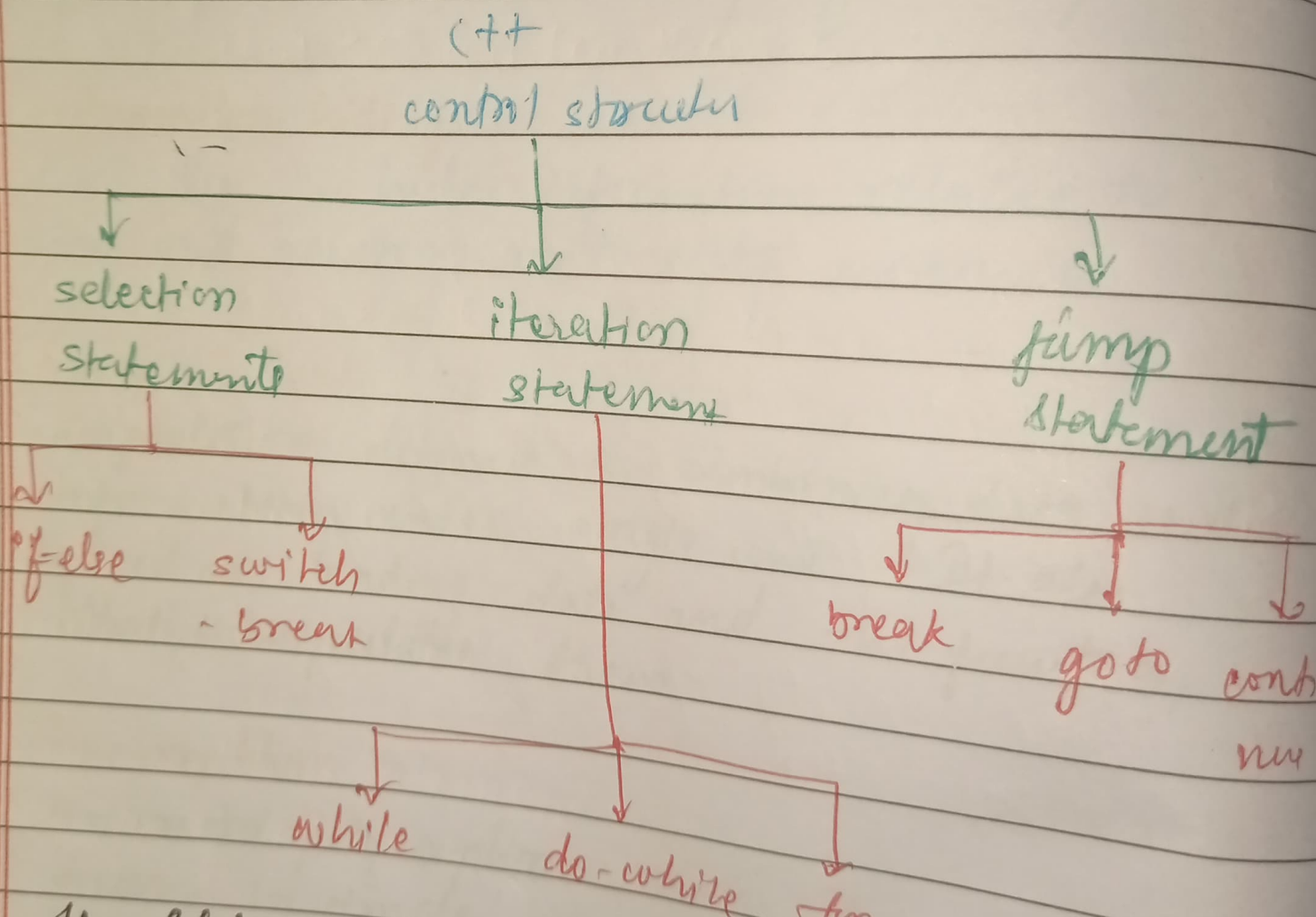
```
int a, b, c;
```

Prabodh
PAGE No. :
DATE :

control Structure in C++

we may counter problem for running same process again and again so for that we use control structure.

control structure is classified in three groups



If - Statements

```
if (expression)
 {
    statement ;
```

class and object

Classes

Class are created using keyword class. A
declaration defines new type of data that
code and data. This used to declare obj
that class

let us see its syntax.

```
class class name {
    private data function
access specifier:
    data and function
access specifier:
    data and function.
    //..
access specifier:
    data and function
} object list;
```

here access specifier are
public
private
protected

by default function and data declared within class
is private to that class.

after using public access specifier then function and
data to other part of program

```
#include <iostream>
#include <cstring>
using namespace std;

class employee {
    char name[80];
public:
    void putname(char *n);
    void getname(char *n);
private:
    double wage;
public:
    void putwage(double w);
    double getwage();
};

void employee::putname(char *n)
{
    strcpy(name, n);
}

void employee::getname(char *n)
{
    strcpy(n, name);
}

void employee::putwage(double w)
{
    wage = w;
}
double employee::getwage()
{
    return wage;
}
```

```
int main()
{
    employee ted;
    char name[80];

    ted.putname("Ted Jones");
    ted.putwage(75000);
    ted.getname(name);
    cout<< name << " makes $";
    cout<< ted.getwage() <<"per yeas.";

    return 0;
}
```

output: Ted Jones makes $ 75000 per year.

we used public access specifies two times we can use that anytime we want to use.
but some programmes use all access specifiers just once.

```
class employee {
  char name[80];
  double wage;
public:
  void putname(char *n)
  void getname(char *n)
  void putwage(double w)
  double getwage();
};
```

same for previous code